

# PROGRAMAÇÃO WEB

UniSenac Campus Pelotas

Escola de Tecnologia da Informação

Prof. Edécio Fernando Iepsen

## Exibindo dados

JSX permite que você coloque marcações em JavaScript. As chaves permitem que você “escape de volta” para o JavaScript para que possa incorporar alguma variável de seu código e exibi-la ao usuário. Por exemplo, isso exibirá `user.name`:

```
return (  
  <h1>  
    {user.name}  
  </h1>  
);
```

Você também pode “escapar para o JavaScript” dos atributos JSX, mas precisa usar chaves *em vez de* aspas. Por exemplo, `className="avatar"` passa a `"avatar"` string como a classe CSS, mas `src={user.imageUrl}` lê o `user.imageUrl` valor da variável JavaScript e passa esse valor como o `src` atributo:

```
return (  
  <img  
    className="avatar"  
    src={user.imageUrl}  
  />  
);
```

Você também pode colocar expressões mais complexas dentro das chaves JSX, por exemplo, concatenação de strings :

#### App.js

Download Redefinir Forquilha

```
1  const user = {
2    name: 'Hedy Lamarr',
3    imageUrl: 'https://i.imgur.com/yX0vd0Ss.jpg',
4    imageSize: 90,
5  };
6
7  export default function Profile() {
8    return (
9      <>
10     <h1>{user.name}</h1>
11     <img
12       className="avatar"
13       src={user.imageUrl}
14       alt={'Photo of ' + user.name}
15       style={{
16         width: user.imageSize,
17         height: user.imageSize
```

Hedy Lamarr



No exemplo acima, `style={{}}` não é uma sintaxe especial, mas um `{}` objeto regular dentro das `style={ }` chaves JSX. Você pode usar o `style` atributo quando seus estilos dependem de variáveis JavaScript.

# renderização condicional

No React, não há sintaxe especial para escrever condições. Em vez disso, você usará as mesmas técnicas usadas ao escrever código JavaScript regular. Por exemplo, você pode usar uma `if` instrução para incluir JSX condicionalmente:

```
let content;
if (isLoggedIn) {
  content = <AdminPanel />;
} else {
  content = <LoginForm />;
}
return (
  <div>
    {content}
  </div>
);
```

Se você preferir um código mais compacto, pode usar o **operador condicional** `?`. Ao contrário `if` do `JSX`, ele funciona dentro do `JSX`:

```
<div>
  {isLoggedIn ? (
    <AdminPanel />
  ) : (
    <LoginForm />
  )}
</div>
```

Quando você não precisa da `else` ramificação, também pode usar uma **sintaxe lógica** `&&` mais curta:

```
<div>
  {isLoggedIn && <AdminPanel />}
</div>
```

Todas essas abordagens também funcionam para especificar atributos condicionalmente.

# Exemplo: Renderização Condicional e Random()



# Exercício:

